

AMD

ZenDNN User Guide

Publication # 57300	Revision # 4.0
Issue Date January 2023	

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

Trademarks

AMD, the AMD Arrow logo, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

Dolby is a trademark of Dolby Laboratories.

ENERGY STAR is a registered trademark of the U.S. Environmental Protection Agency.

HDMI is a trademark of HDMI Licensing, LLC.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium.

Microsoft, Windows, Windows Vista, and DirectX are registered trademarks of Microsoft Corporation.

MMX is a trademark of Intel Corporation.

OpenCL is a trademark of Apple Inc. used by permission by Khronos.

PCIe is a registered trademark of PCI-Special Interest Group (PCI-SIG).

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

Dolby Laboratories, Inc.

Manufactured under license from Dolby Laboratories.

Rovi Corporation

This device is protected by U.S. patents and other intellectual property rights. The use of Rovi Corporation's copy protection technology in the device must be authorized by Rovi Corporation and is intended for home and other limited pay-per-view uses only, unless otherwise authorized in writing by Rovi Corporation.

Reverse engineering or disassembly is prohibited.

USE OF THIS PRODUCT IN ANY MANNER THAT COMPLIES WITH THE MPEG-2 STANDARD IS EXPRESSLY PROHIBITED WITHOUT A LICENSE UNDER APPLICABLE PATENTS IN THE MPEG-2 PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, L.L.C., 6312 S. FIDDLERS GREEN CIRCLE, SUITE 400E, GREENWOOD VILLAGE, COLORADO 80111.

Contents

Revision History5
Chapter 1 Introduction6
Chapter 2 Scope7
Chapter 3 Release Highlights8
Chapter 4 Supported OS and Compilers9
4.1 OS9
4.2 Compilers9
Chapter 5 Runtime Dependencies10
Chapter 6 Logs11
Chapter 7 License13
Chapter 8 Technical Support14

List of Tables

Table 1.	Log Actors.	11
----------	------------------	----

Revision History

Date	Revision	Description
January 2023	4.0	Updated supported TensorFlow, ONNX Runtime, and PyTorch versions.
June 2022	3.3	<ul style="list-style-type: none">Updated supported TensorFlow and PyTorch versions.Removed Chapter 5 Prerequisites and Chapter 6 AOCC and AOCL (AMD-BLIS) Library Installation.
December 2021	3.2	Updated supported TensorFlow, ONNX Runtime, and PyTorch versions.
August 2021	3.1	Updated supported TensorFlow versions.
April 2021	3.0	Initial version.

Chapter 1 Introduction

ZenDNN (Zen Deep Neural Network) Library accelerates deep learning inference applications on AMD CPUs. This library, which includes APIs for basic neural network building blocks optimized for AMD CPUs, targets deep learning application and framework developers with the goal of improving inference performance on AMD CPUs across a variety of workloads, including computer vision, natural language processing (NLP), and recommender systems. ZenDNN leverages oneDNN/DNNL v2.6.3's basic infrastructure and APIs. ZenDNN optimizes several APIs and adds new APIs, which are currently integrated into TensorFlow, ONNX Runtime, and PyTorch. ZenDNN depends on:

- BLAS-like Library Instantiation Software (AOCL-BLIS) library for its BLAS (Basic Linear Algebra Subprograms) API needs
- AMD Math Library (LibM) for Core Math needs
- Composable Kernel for convolutions using an implicit GEMM algorithm

AOCL-BLIS and AOCL-LibM are required dependencies for ZenDNN, whereas AMD Composable Kernel is an optional dependency.

Chapter 2 Scope

The scope of ZenDNN is to support AMD EPYC™ CPUs on the Linux® platform. ZenDNN v4.0 offers optimized primitives, such as Convolution, MatMul, Elementwise, and Pool (Max and Average) that improve performance of many convolutional neural networks, recurrent neural networks, transformer-based models, and recommender system models. For the primitives not supported by ZenDNN, execution will fall back to the native path of the framework.

Chapter 3 Release Highlights

Following are the highlights of this release:

- ZenDNN library is integrated with TensorFlow v2.10, ONNX Runtime v1.12.1, and PyTorch v1.12.
- Python v3.7-v3.10 have been used to generate the following wheel files (*.whl):
 - TensorFlow v2.10
 - PyTorch v1.12
 - ONNX Runtime v1.12.1
- Added the following environment variables for tuning performance:
 - Memory Pooling (Persistent Memory Caching):
 - ZENDNN_ENABLE_MEMPOOL for all the TensorFlow models
 - Added MEMPOOL support for INT8 models
 - Convolution Operation:
 - ZENDNN_CONV_ALGO for all the TensorFlow models
 - Added new ALGO paths
 - Matrix Multiplication Operation:
 - ZENDNN_GEMM_ALGO for all the models
 - Added new ALGO paths and experimental version of auto-tuner
- NHWC (default format) and Blocked Format (NCHWc8) continue to be supported.

ZenDNN library is intended to be used in conjunction with the frameworks mentioned above and cannot be used independently. It is inherited from oneDNN v2.6.3.

The latest information on the ZenDNN release and installers is available on AMD Developer Central (<https://www.amd.com/en/developer/zendnn.html>).

Chapter 4 Supported OS and Compilers

This release of ZenDNN supports the following Operating Systems (OS) and compilers:

4.1 OS

- Ubuntu[®] 20.04 LTS and later
- Red Hat[®] Enterprise Linux[®] (RHEL) 9.0 and later
- CentOS Stream 9 and later

4.2 Compilers

GCC 9.3 and later

Chapter 5 Runtime Dependencies

ZenDNN has the following runtime dependencies:

- GNU C library (*glibc.so*)
- GNU Standard C++ library (*libstdc++.so*)
- Dynamic linking library (*libdl.so*)
- POSIX Thread library (*libpthread.so*)
- C Math Library (*libm.so*)
- OpenMP (*libomp.so*)
- Python v3.7-v3.10 for:
 - TensorFlow v2.10
 - ONNX Runtime v1.12.1
 - PyTorch v1.12

Since ZenDNN is configured to use OpenMP, a C++ compiler with OpenMP 2.0 or later is required for runtime execution.

Chapter 6 Logs

Logging is disabled in the ZenDNN library by default. It can be enabled using the environment variable **ZENDNN_LOG_OPTS** before running any tests. Logging behavior can be specified by setting the environment variable **ZENDNN_LOG_OPTS** to a comma-delimited list of **ACTOR:DBGLVL** pairs.

The different ACTORS are as follows:

Table 1. Log Actors

Actor	Description
ALGO	Logs all the executed algorithms.
CORE	Logs all the core ZenDNN library operations.
API	Logs all the ZenDNN API calls.
TEST	Logs all the calls used in API tests, functionality tests, and regression tests.
PROF	Logs the performance of operations in millisecond.
FWK	Logs all the framework (Tensorflow, ONNX Runtime, and PyTorch) specific calls.

For example:

- To turn on info logging, use **ZENDNN_LOG_OPTS=ALL:2**
- To turn off all logging, use **ZENDNN_LOG_OPTS=ALL:-1**
- To only log errors, use **ZENDNN_LOG_OPTS=ALL:0**
- To only log info for ALGO, use **ZENDNN_LOG_OPTS=ALL:-1,ALGO:2**
- To only log info for CORE, use **ZENDNN_LOG_OPTS=ALL:-1,CORE:2**
- To only log info for API, use **ZENDNN_LOG_OPTS=ALL:-1,API:2**
- To only log info for PROF (profile), use **ZENDNN_LOG_OPTS=ALL:-1,PROF:2**
- To only log info for FWK, use **ZENDNN_LOG_OPTS=ALL:-1,FWK:2**

The Different Debug Levels (DBGLVL) are as follows:

```
enum LogLevel
{
    LOG_LEVEL_DISABLED = -1,
    LOG_LEVEL_ERROR = 0,
    LOG_LEVEL_WARNING = 1,
    LOG_LEVEL_INFO = 2,
    LOG_LEVEL_VERBOSE0 = 3,
    LOG_LEVEL_VERBOSE1 = 4,
    LOG_LEVEL_VERBOSE2 = 5
};
```

Chapter 7 License

ZenDNN is licensed under Apache License Version 2.0. Refer to the “LICENSE” file for the full license text and copyright notice.

This distribution includes third party software governed by separate license terms.

3-clause BSD license:

- Xbyak (<https://github.com/herumi/xbyak>)
- Googletest (<https://github.com/google/googletest>)
- Instrumentation and Tracing Technology API (<https://github.com/intel/ittapi>)

Apache License Version 2.0:

- oneDNN (<https://github.com/oneapi-src/oneDNN>)
- Xbyak_aarch64 (https://github.com/fujitsu/xbyak_aarch64)
- TensorFlow (<https://github.com/tensorflow/tensorflow>)

Boost Software License, Version 1.0:

Boost C++ Libraries (<https://www.boost.org/>)

BSD/Apache/Software Licenses from PyTorch:

PyTorch (<https://github.com/pytorch/pytorch>)

This third-party software, even if included with the distribution of the Advanced Micro Devices software, may be governed by separate license terms, including without limitation, third-party license terms, and open-source software license terms. These separate license terms govern use of the third-party programs as set forth in the THIRD-PARTY-PROGRAMS file.

Chapter 8 Technical Support

Please email zendnnsupport@amd.com for questions, issues, and feedback on ZenDNN.